Seamless access to healthcare folders with strong privacy guarantees

Tristan Allard, Nicolas Anciaux, Luc Bouganim, Philippe Pucheral, Romuald Thion

▶ **To cite this version:**

Tristan Allard, Nicolas Anciaux, Luc Bouganim, Philippe Pucheral, Romuald Thion. Seamless access to healthcare folders with strong privacy guarantees. International Journal of Healthcare Delivery Reform Initiatives, 2009, 1 (4), pp.82-107. hal-00623899

HAL Id: hal-00623899
https://hal.science/hal-00623899

Submitted on 15 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Seamless Access to Healthcare Folders with Strong Privacy Guarantees

Tristan Allard, University of Versailles & INRIA Rocquencourt, France
Nicolas Anciaux, INRIA Rocquencourt, France
Luc Bouganim, INRIA Rocquencourt, France
Philippe Pucheral, University of Versailles & INRIA Rocquencourt, France
Romuald Thion, INRIA Grenoble, France

ABSTRACT

During the past decade, many countries launched ambitious Electronic Health Record (EHR) programs with the objective to increase the quality of care while decreasing its cost. Pervasive healthcare aims itself at making healthcare information securely available anywhere and anytime, even in disconnected environments (e.g., at patient home). Current server-based EHR solutions badly tackle disconnected situations and fail in providing ultimate security guarantees for the patients. The solution proposed in this paper capitalizes on a new hardware device combining a secure microcontroller (similar to a smart card chip) with a large external Flash memory on a USB key form factor. Embedding the patient folder as well as a database system and a web server in such a device gives the opportunity to manage securely a healthcare folder in complete autonomy. This paper proposes also a new way of personalizing access control policies to meet patient's privacy concerns with minimal assistance of practitioners. While both proposals are orthogonal, their integration in the same infrastructure allows building trustworthy pervasive healthcare folders.

Keywords: Access Control; Anonymization; Availability; Electronic Health Record; Keys; Privacy; Privacy Preserving Data Publishing; Secure Chip; Synchronization; Usage Control

1. INTRODUCTION

Driven by the need to improve the quality of care while decreasing costs, many countries around the world are setting up large scale Electronic Health Record (EHR) systems gathering the medical history of individuals. Interoperability among heterogeneous healthcare information systems and privacy preservation are two main challenges in this context. Pervasive healthcare on its side strive to remove location and time constraints to access patient's healthcare folders. Cares provided at home to elderly or disabled people illustrate well the need for pervasiveness. In this context healthcare data is often collected and consulted at home by practitioners having different privileges and acting at different time periods. Healthcare information must be safely exchanged among practitioners to improve care coordination but no connection to the Internet can be always guaranteed. Data can also be issued by institutions external to the care coordination (e.g., a medical lab) and join the patient's folder. The folder is sometimes accessed by practitioners outside patient's home (e.g., doctor's office, hospital). Finally, a large collection of folders can be targeted by epidemiological studies for a general health benefit. In this paper, we discuss how smart objects can provide a seamless access to healthcare folders without privacy breach in all these situations.

EHR systems aim at answering most of the requirements mentioned above. The objective of centralizing medical information in database systems is manifold[1]: completeness (i.e., to make the information complete and up to date), availability (to make it accessible through the internet 24h-7 days a week), usability (to organize the data and make it easily queryable and interpretable), consistency (to guarantee integrity constraints and enforce atomicity and isolation of updates) and durability (to protect the data against failure). A recent report identified more than 100 EHR running projects worldwide at the scale of a country or regions in 2007 (Door, 2008). Other reports suggest that about 25% of U.S. healthcare practices use EHR systems. Within Europe these figures vary greatly between countries, from 15% in Greece up to 90% in the Netherlands today.

Regarding pervasiveness, healthcare folders can be reached by allowing internet connections to the server(s) through mobile devices (e.g., laptop, PDA, tablet PC). This however requires that every point of the territory be connected through a secure, fast, reliable and cheap network, a situation uncommon in many countries and regions today.

In addition, and despite the unquestionable benefit of EHR systems in terms of quality of care, studies conducted in different countries show that patients are reluctant to use existing EHR systems arguing increasing threats on individual privacy (The Times, 2008; The International Council on Medical & Care Compunetics, 2009). This suspicion is fueled by computer security surveys pointing out the vulnerability of database servers against external and internal attacks (Gordon et al., 2006). Indeed, centralizing and organizing the information make it more valuable, thereby motivating attacks and facilitating abusive usages. Regardless of the legislation protecting the usage of medical data and of the security procedures put in place at the servers, the patient has the sense of losing control over her data.

Hence, implementing a trustworthy pervasive access to healthcare folders requires addressing accurately the following issues:

1. How to make patient's healthcare folder available in a disconnected mode?
2. How to make patient's healthcare folder seamlessly available in a connected area?
3. How to make the patient trust the EHR security?
4. How to get the patient consent about a pervasive use of her healthcare folder?

As discussed above, existing EHR systems answer well issue 2 but fail in answering issue 1 and issue 3. Therefore, EHR systems fail also in answering issue 4 precisely due to the lack of server trustworthiness.

This paper suggests a new way of organizing EHR to address issues 1 to 4 all at once. The solution proposed capitalizes on a new hardware device called Secure Portable Token (SPT) hereafter. Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized) on a USB key form factor (Eurosmart, 2008). A SPT can host on-board data and run on-board code with proven security properties. Embedding a database system and a web server in a SPT gives the opportunity to manage securely a healthcare folder in complete autonomy. Accessing the on-board folder at patient's home requires a simple rendering device (e.g., a netbook or PDA) equipped with an USB port and running a web browser. Then issue 1 is tackled by construction. The SPT security properties

(tamper-resistant hardware, certified embedded software) answer issue 3 with a much higher confidence than any traditional server can provide.

Issue 2 becomes however more difficult to address. Indeed, the patient folder cannot be accessed without being physically in possession of the patient's SPT. We propose two complementary solutions to tackle situations where remote accesses to the folder are mandatory. The first solution is to reintroduce a server in the architecture so that a secure exchange of information can be organized between the patient and a trusted circle of persons (e.g., the family doctor expressing her opinion in an emergency situation without having the patient's folder on hand). The solution is such that patient's data is never stored in the clear on the server and encryption/decryption keys are known only by the SPT participating to the trusted circle of person defined by the patient. The second solution answers Privacy Preserving Data Publishing (PPDP) requirements (Fung, 2009), where patient's data must be made available to epidemiological studies. We propose a secure publishing mechanism such that a global anonymized view of data collected from different folders can be produced without disclosing any link to individuals. These two solutions together tackle issue 2 without compromising issue 3.

One may consider that answering issue 3 leads to answer issue 4 as well. This is unfortunately not true. Trusting the EHR security is a prerequisite for the patient to give her consent about a pervasive use of her medical folder but it is definitely not a sufficient condition. Expressing an enlightened consent means understanding and accepting an access control policy specifying who (individuals or roles) is granted access to which part of her folder. The high number of people interacting with the folder, the diversity of their roles, the complexity of the medical information and the intrinsic difficulty to determine which data (or data association) reveals a given pathology makes this objective highly difficult to reach. This paper proposes a new and pragmatic alternative to define access control policies manageable by a patient with minimal assistance of practitioners. This solution complements well the SPT-based EHR architecture by answering issue 4. This solution is however orthogonal to the SPT-based architecture and we believe it could apply to many healthcare information systems.

As a conclusion, the objective of this paper is twofold. First, it discusses to which extent existing EHR architectures can meet the four requirements introduced above (Section 2) and proposes an alternative based on the SPT hardware device (Sections 3 and 4). Second, it discusses whether existing access control policies and confidentiality mechanisms can capture the patient's consent (Section 5) and proposes a solution relying on a new access control model (Section 6). Section 7 relates an experimentation in the field of a pervasive EHR architecture combining both proposals, in the context of home care provided to elderly people.

2. BACKGROUND

An electronic health record system is a collection of electronic patient folders, each containing the complete medical history of an individual, managed and consulted by authorized health professionals across several organizations (Alliance, 2007). Building an EHR system requires interconnecting various heterogeneous health information systems of disparate organizations in order to aggregate the medical data they maintain separately (e.g., hospitals, practitioners and pharmacists data related to a same individual).

Hence, the first challenge tackled by EHR programs is providing interoperability between heterogeneous systems. As pointed out in the introduction, ensuring data availability even in disconnected environments and enforcing data security are two additional and mandatory challenges. The next subsections present a state of the art on these three challenges.

2.1. EHR Interoperability

Three main approaches can be distinguished according to the level of integration targeted between existing health information systems.

The first approach consists in interconnecting existing autonomous systems in a wider infrastructure with no data centralization and a minimal central control. The Danish Healthcare Data Network (Pedersen, 2006)[2] is representative of this category. It connects the already secure intranets of care organizations via VPNs over the Internet, progressively from organizations to counties, counties to regions, and regions to nation. The Danish EHR effort has mainly consisted in defining a common data model representing clinical data. The USA have adopted a federal approach to build the EHR. At the region scale, Regional Health Information Organizations (RHIOs) are multi-stakeholder organizations that enable the exchange of health information between local health organizations (e.g., CalRHIO for the Californian RHIO). At the nation scale, the Nationwide Health Information Network (NHIN) project, supervised by the Office of the National Coordinator for Health IT (ONC), aims at enabling secure health information exchange across the USA by using RHIOs as regional building blocks. The NHIN will be a "network of networks" built over the Internet.

The second approach strengthens the integration thanks to centralized indexes and/or data summaries. The National Health Society in the United Kingdom has launched the Care Record Service (CRS) project. First, CRS aims at linking together the existing Electronic Medical Record (EMR) of an individual, thus constituting a virtual unique health folder. Navigating between the EMRs of an individual and gathering detailed data will be easier; furthermore, by sharing data across EMRs, duplication of - e.g., administrative - data will be useless. Second, CRS aims at storing summaries of detailed data on the "Spine", an already existing central system currently in charge of delivering health related services (e.g., ePresciptions, eReservations). The Secondary Uses Service (SUS) of CRS will use summarized data to draw reports and analysis about collected care information. With its two-level functional architecture, the Diraya project from Andalusia is similar to the CRS UK project. First, detailed data in EMRs are kept where they are produced (e.g., hospitals) and the central Diraya system indexes them. Second, Diraya centralizes what is called the "main data", that is the data most frequently accessed. In the Netherlands, the National Healthcare Information Hub project (LSP in Dutch), lead by Nictiz is basically a central index storing the location of every individual's EMRs. The Austrian ELGA initiative (Husek, 2008) is similar to the LSP project. The Canadian national program Infoway-Inforoute funds provincial EHR projects, most of these focusing on interoperability between care organizations. For example, the Alberta Netcare EHR centralizes regionally the patients' summary data. The Yukon Telehealth project makes local EMRs accessible to remote specialist practitioners.

The most integrated approach seeks to gather all EMRs related to the same individual into a centralized healthcare folder. In the United States, some private organizations had already felt

the need to aggregate all their patients' data in a single folder before the coming of RHIOs. For example, the Veteran Administration Medical Center developed the VistA system (Brown et al, 2003), a Health Information System (HIS) enabling health centers with VistA to share their patients' data. The French national program Dossier Medical Personnel (DMP) aims also at centralizing healthcare folders hosted by selected database service providers. In another spirit, systems like Google Health™ and Microsoft's HealthVault™ propose individuals to centralize their Personal Health Records (PHRs) on their own initiative. Both load medical data directly from the patient's health centers that do agree, offer practical tools for individuals (e.g., drug interactions, hospitals searches), and can provide a controlled access to the PHR to a selected set of persons. Both are free and the users are simply asked to trust their privacy policy.

2.2. EHR Availability

All EHR systems mentioned above provide a 24h/7day a week availability assuming the servers are active and an internet connection can be established to reach them. This is unfortunately not the case in every place and every situation, introducing the need for disconnected accesses to healthcare folders.

In the United Kingdom, the Health eCard is a private initiative that proposes to store encrypted copies of full EMRs in specifically designed smart cards, making patients' health data available in disconnected situations (e.g., emergency situations, home consultation).

The German organization Gematik leads the eGK, an ambitious project mixing a traditional infrastructure with smart cards in order to tackle connected and disconnected situations (Smart Card Alliance-b, 2006). Both patients and professionals are equipped with a smart card, patient smart cards storing EHRs while professional smart cards are being used for strong authentication, digital signature and encryption/decryption of documents. The infrastructure holds a centralized copy of the EHRs, accessible through the internet. This project is still at a preliminary stage.

In the USA, many private initiatives issued by care centers tackle the "disconnected access" requirement (Smart Card Alliance-a, 2006), e.g., the University of Pittsburgh Medical Center Health Passport Project (HPP), the Florida eLife-Card, Queens Health Network, Mount Sinai Medical Center Personal Health Card. All of them store a copy of critical health information encrypted on a smart card to make it available in case of emergency.

Taiwan has launched in 2001 a project to replace the traditional paper health cards by smart cards (Smart Card Alliance, 2005). Smart cards are used exactly as paper cards were used. They permanently store administrative personal and summary health data, and temporarily store the medical data related to the last six visits. Every six visits, the temporary medical data are uploaded into the Taiwanese health infrastructure. The smart card health project is seamlessly integrated with the previous health infrastructure, providing a strong patient authentication and a paperless data management.

While many initiatives tackle the disconnected access challenge, the low storage capacities of the smart cards used in the aforementioned projects (i.e., at best a few hundreds of kilobytes) severely limit the quantity of on-board data, and then the benefit of the approach.

2.3. EHR Security

Strong authentication is usually required to connect to EHR servers. Health professionals authenticate with a smart-card (e.g., the CRS in UK, the LSP in the Netherlands), as well as patients accessing to their medical folder (e.g., the Diraya initiative in Andalusia). In addition, communication channels can be protected by cryptographic techniques, based on protocols such as TLS (Internet Engineering Task Force, 2008), enabling entities to securely exchange messages (i.e., encryption, integrity protection, non repudiation of messages), and security measures are implemented on central servers. However, this is not sufficient to put trust in the system.

The suspicion is fueled by computer security surveys pointing out the vulnerability of database servers against external and internal attacks (Gordon et al., 2006). Database systems are identified as the primary target of computer criminality (Gordon et al., 2006), and even the most well defended servers, including those of Pentagon (The Financial Times, 2007; Liebert, 2008), FBI (The Washington Post, 2007) and NASA (Computer World, 2003), have been successfully attacked. In addition, nearly half of the attacks (Gordon et al, 2006) come from the inside (employees) of the companies or organizations. In addition, there are many examples where negligence leads to personal data leakages. To cite a few, thousands of Medicare and Medicaid patients in eight states have been lost in a HCA regional office (FierceHealthIT news, 2006) and Hospitals County published by accident medical records on the web (FierceHealthIT news, 2008; WFTV, 2008) including doctors' notes, diagnoses, medical procedures and possibly names and ages of patients. A recent study shows that 81% of US firms declare loosing employees laptops with sensitive data (Computer World, 2006). Data loss is so frequent that a research project called DataLossDB has been created to report such incidents.

In practice, EHRs are thus very difficult to protect. This legitimates the reserves expressed by both practitioners and patients about EHR programs (The Times, 2008; eHealth Insider, 2008). In the Netherlands, privacy and access concerns are major arguments for the postponement of the national EHR (The International Council on Medical & Care Compunetics, 2009). In particular, the lack of security measures limiting data access for service providers and the loss of control on their own data has been identified as a main reason for citizens to opt-out of the system.

Only EMRs stored in personal and secure hardware such as smart-cards (see Section 2.2) can benefit from true privacy enforcement. However, (1) the storage capacity of the smart-cards used by current projects (e.g., from KB to MB) is too low to store a complete EHR, limiting the data availability in disconnected situations, (2) their low connectivity makes the hosted data seldom available, and (3) their portable nature makes them subjects to losses or destruction. Moreover, to tackle availability, these projects rely on central servers unable to enforce the smart-cards security level (Eurosmart, 2008).

We believe that both data privacy and full availability can be achieved altogether. The solution we propose is centered on a personal smart-card based device storing (the most significant part of) the patient folder and extending the security sphere of its secure hardware to traditional central servers.

3. A SECURE, PORTABLE MEDICAL FOLDER

Researches conducted in the PlugDB[3] project led us to design a lightweight Database Management System (DBMS) embedded in a new form of tamper-resistant token, called hereafter Secure Portable Token (SPT). Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized) on a USB key form factor (Eurosmart, 2008). A SPT can host a large volume of on-board data and run on-board code with proven security properties thanks to its tamper-resistant hardware and a certified operating system (Eurosmart, 2008). The main target of the PlugDB technology is the management of secure and portable personal folder. Healthcare folders are very good representative of large personal folders where security and portability are highly required.

Compared to smart cards used in other EHR projects (see Section 2), the storage capacity of a SPT is roughly four orders of magnitude higher. Henceforth, this makes sense to embed the whole patient folder in her SPT and make it available in disconnected mode. In addition to the data, a complete chain of software is embedded in the SPT microcontroller: (1) a Web server, (2) servlets implementing the application, (3) a JDBC bridge, (4) a DBMS engine managing the on-board database and enforcing access control. Hence, the SPT along with its embedded database and software can be seen as a full-fledged server accessible through any web browser running on any device equipped with a USB port (e.g., laptop, tablet-PC, PDA and even cell-phone). Compared to a regular server, the SPT server is personal, pluggable, does not require any network connection and provides unprecedented security guarantees.

The specific hardware architecture of the SPT introduces however many technical challenges. We detail below the most important ones.

3.1. SPT Hardware and Operating System

A SPT combines in the same hardware platform a secure chip and a mass storage NAND FLASH memory (several Gigabytes soon). The secure chip is of the smart card type, with a 32 bit RISC CPU clocked at about 50 MHz, memory modules composed of ROM, tens of KB of static RAM, a small quantity of internal stable storage (NOR FLASH) and security modules. The mass storage NAND FLASH memory is outside the secure chip, connected to it by a bus, and does not benefit from the chip hardware protection.

Gemalto, the smart card world leader has developed an experimental SPT platform. This platform includes a new multi-tasking operating system allowing the development of Web applications based on JAVA and Servlet technology, and thus offering a standardized means to integrate services or embedded Web applications to the SPT. The operating system supports natively: the USB 2.0 protocol and the internet protocol IP for communicating with the external world (Vandewalle, 2004); multi-threaded Java applications; cryptographic primitives (some of which implemented in hardware); memory management and garbage collection; Servlet management and Web server. For more technical details on the hardware platform and the operating system, we refer the reader to (http://www-smis.inria.fr/~DMSP).

3.2. Embedded Database System

DBMS designers have produced light versions of their systems for personal assistants (e.g. Oracle-lite, DB2 everyplace, SQLServer for Window CE) but they never addressed the more complex problem of embedding a DBMS in a chip. Initial attempts towards a smart card DBMS was ISOL's SQLJava Machine (Carrasco, 1999), the ISO standard SCQL (ISO/IEC, 1999) and the MasterCard Open Data Storage (MasterCard International, 2002). All these proposals concerned traditional smart cards with few resources and therefore proposed basic data management functionalities (close to sequential files). Managing embedded medical folders requires much more powerful storage, indexation, access control and query capabilities. PicoDBMS was the first full fledged relational DBMS embedded in a smart card (Pucheral et al, 2001) and was implemented on top of Gemalto's smart card prototypes (Anciaux et al., 2001). PicoDBMS has been designed for managing databases stored in a (Megabyte sized) EEPROM stable memory integrated in the secure chip and protected by the chip tamper-resistance.

The SPT framework introduces important new challenges (Anciaux et al., 2007):

1. How to support complex queries over a large on-board database (Gigabyte sized) with very little RAM (a few kilobytes)?
2. How to organize the data storage and the indexes with an acceptable insert/update time considering the peculiarities of NAND Flash memory (fast reads, costly writes, block-erase-before-page-rewrite constraint)?
3. How to protect the on-board database against confidentiality and integrity attacks (the external Flash being not hardware protected) while keeping acceptable query performance?

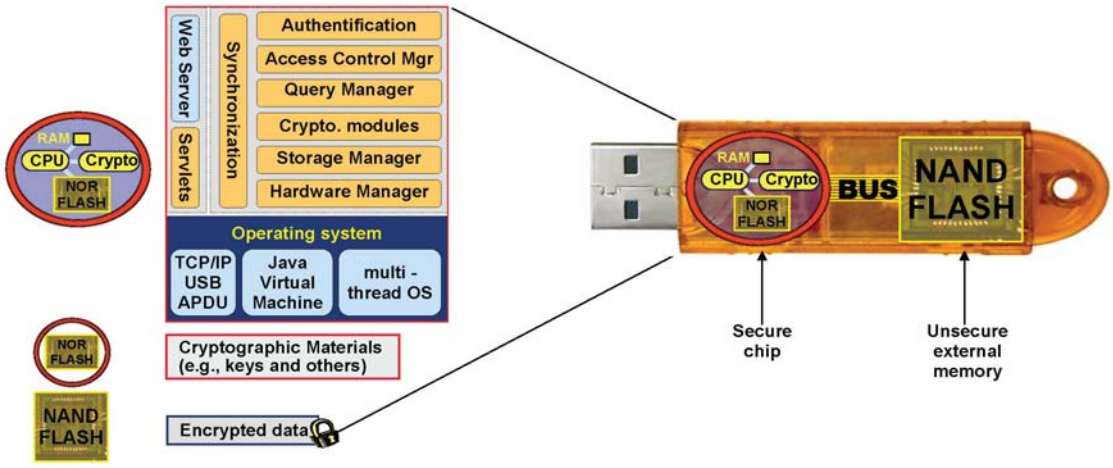

Figure 1. Secure Portable Token

The SPT architecture and the organization of the embedded software components are illustrated in Figure 1. The on-board code and sensitive data (e.g., cryptographic keys) reside in the secure chip; patient's data reside in the insecure external memory, previously encrypted by the secure execution environment. We detail below the components related to the technical challenges mentioned above.

The Query Manager is in charge of parsing the incoming database queries, building an optimal query execution plan and executing it. This module must consider peculiar execution strategies to answer complex SQL queries over a large quantity of data with little RAM (challenge 1). To tackle this challenge, we designed a massive indexing scheme presented in (Anciaux, Benzine, et al, 2007), which allows processing complex queries while consuming as little RAM as possible and still exhibiting acceptable performance. The idea is to combine in the same indexing model generalized join indices and multi-table selection indices in such a way that any combination of selection and join predicates can be evaluated by set operations over lists of sorted tuple identifiers. The operator library (algorithms for the operators of the relational algebra, e.g., select, project, join and aggregate) and the execution engine integrate those techniques.

The Storage Manager on which the query manager relies to access the database content (index and tables) is directly concerned with challenge 2. Indeed, the proposed massive indexation scheme causes a difficult problem in terms of Flash updates, due to the severe read/write constraints of NAND Flash (rewriting NAND Flash pages is a very costly operation). Therefore, we designed a structure which manages data and index keys sequentially so that the number of rewrites in Flash is minimized. The use of summarization structures based on bloom filters (Bloom, 1970) and vertical partitioning reduce the cost of index lookups (Yin et al, 2009). These additional structures are also managed in sequence. A first implementation of this principle has been patented jointly by INRIA and Gemalto (Pucheral & Yin, 2007) and is integrated in the current DBMS prototype.

The Hardware Manager embeds the methods for accessing the different memory modules of the SPT. It includes techniques associated with challenge 3 to protect the confidentiality and the integrity of the data, in an efficient way with respect to DBMS access patterns. Indeed, our massive indexation technique leads to numerous, random and fine grain accesses to raw data. We conducted preliminary studies (Anciaux et al., 2006), in which we combine encryption, hashing and timestamping techniques with query execution techniques in order to satisfy three conflicting objectives: efficiency, high security and compliance with the chip hardware resources.

The Access Control Manager is in charge of enforcing the access control policy defined to grant/deny access to pieces of the patient folder to the current user. Privileges can be associated to individual users or roles. To help collecting patients' consent, each patient should be given the chance to personalize a predefined access control policy. Hence, the Access Control Manager plays an important role in answering challenge 4 identified in the introduction. Access control is more deeply discussed in Section 6.

3.3. Data Availability and Security

Any terminal equipped with an USB port and a Web Browser can interact with the SPT and get the data he is granted access to. Hence, when no Internet connection is available (e.g., emergency situations, home intervention, remote server breakdown) SPTs guarantee patients' data availability, thereby achieving challenge 1 identified in the introduction. Furthermore, local connections to SPTs do not suffer from unpredictable performance due to overloaded remote servers or low quality connections: the embedded server is mono-user and USB-2 communication throughput is guaranteed.

In terms of security, patient's data resides in the external NAND Flash memory. As stated in section 3.2, this memory is not hardware protected so that its content must be encrypted to prevent confidentiality attacks and hashed to prevent integrity attacks. The cryptographic keys serving this purpose reside in the NOR Flash memory and are protected by the tamper-resistance of the secure chip. The encryption/decryption/hashing processes physically take place in the secure chip and are similarly hardware protected (i.e., see the red circle of Figure 1). More generally, the complete software chain (web server, servlets, DBMS) runs in the secure chip and benefits from its tamper-resistance. Hence, the authentication, access control, and query steps are all hardware protected. The security of the complete architecture thereby relies on the tamper-resistance of the secure chip. The security of our hardware platform and of the embedded code is under certification with the goal to reach the highest security level ($EAL4^+$), usually required for smart card chips used in the medical domain. This makes attacks highly improbable and extremely costly to conduct. Considering that the security of a system lies in the fact that the cost to conduct an attack outweighs its benefit, the security of our architecture is reinforced by the extreme cost of attacks and their small benefit (disclosure of a single patient's folder). Consequently, we argue that our architecture achieves convincingly challenge 4 identified in the introduction.

4. A SECURE, PERVASIVE MEDICAL FOLDER

Section 3 tackled challenges 1 and 3; this section focuses on challenge 2 that is making patient's healthcare folder seamlessly available in a connected area. Solving this challenge allows answering questions like: (1) how can the family doctor express her opinion about an emergency situation without the Patient's SPT on hand, or (2) how can survey institutes draw useful data analysis without having access to the patients' raw data? We introduce a central server in the architecture as a mean to obtain the required data availability.

This must be achieved without losing the benefits of the SPT in terms of security and control by its owner. This entails never revealing sensitive information to the central server. Two rules arise from this statement: (1) sensitive data must be stored encrypted on the server storage media, and (2) sensitive data must be encrypted and decrypted in a secure execution environment, i.e., a SPT.

To secure the communications, we use protocols such as TLS (Internet Engineering Task Force, 2008). TLS relies on a certificate authority to emit trusted certificates linking an identity to a public key. The Professionals, patients and central server safely communicate after having exchanged and checked their respective certificates. Certificates are inserted in the SPTs' secure internal memory at the beginning of their life cycle, before being delivered to its owner. The server is in charge of securing his own certificate. Note that SPTs are not durable: they may be lost or broken. Hence, to make certificates and pairs of (public key, private key) durable, they must be replicated in a Trusted Third Party (TTP). We do not detail those protocols further in this paper.

In this section, we first classify data according to their needs in terms of privacy and depict an SPT-centered architecture fulfilling these needs. Second, we focus on synchronization issues between the patient's SPT, the central server, and external entities (e.g., laboratories). Third, we present the Privacy Preserving Data Publishing protocol. Finally, we describe a comprehensive use case of the system.

4.1. A Data Classification Driven by Privacy Concerns

We call *Secret Data* (SD) the pieces of information the patient considers so sensitive (e.g., psychological analysis) that he cannot accept to delegate their storage to a remote server. Secret data remains confined to the patient's SPT. Hence SD durability is under the patient responsibility and SD availability requires the presence of the patient's SPT.

We call *Regular Data* (RD) the pieces of information the patient consents to replicate on a remote server in the clear. Such data is protected by the security policy enforced by the server and can be accessed on line by any practitioner having the required privileges. The access control policy enforced by the server and the SPT is assumed to be identical. However, the server does not benefit from the tamper-resistance of the SPT and RD can be accessed on the server without prior patient knowledge. What is actually considered as RD depends on the patient feeling, e.g., administrative data, non-sensitive medications (e.g., aspirin), non sensitive diagnosis (e.g., flue). Being replicated on the server, RD benefits from the server on-line availability and durability properties.

We call *Confined Data* (CD) the pieces of information which are too sensitive to be managed as RD but for which on-line availability and/or durability is mandatory for care practice (e.g., HIV diagnosis, chemotherapy medication, MRI image). CD is replicated encrypted on the server but the encryption keys are never present at the server. Hence CD is protected against server attacks. Encryption keys are stored and managed by SPT devices only. To ensure on-line availability, encryption keys can be shared by the SPTs of a selected set of persons, named hereafter *trusted circle* (e.g., the family doctor and some specialist physicians). Members of the trusted circle can access CD on the server and their SPT can decrypt them. Defining trusted circles is up to the patient. Durability is guaranteed by the server similarly to clear-text data. However, recovering CD entails recovering the related encryption keys. This can be achieved either by a pass phrase or by a TTP registering encryption keys. Note that Secret Data can be made durable by declaring them as confined, without sharing the encryption keys and assuming the patient trusts the cryptographic protocols.

Finally, we call *Anonymous Data* (AD) the pieces of information the patient agrees to externalize in order to contribute to a health survey (e.g., epidemiological study), provided these information will be properly anonymized. AD can be an extraction of – or a logical view built from – any data present in the folder (either SD, CD or RD). To preserve anonymity, AD can not be exported directly by the SPT hosting it but must go through a specific Privacy Preserving Data Publishing protocol.

Data classification is under patient responsibility, possibly with external help (e.g., his family doctor). The patient can change his mind afterwards (e.g., following the advice of his doctor) according to the following hierarchy: secret data → confined data → regular data → anonymous data. Any other change is uncertain, e.g., when changing from regular data to secret data, the clear regular data could have been queried or copied beforehand; the patient cannot be sure of its secrecy.

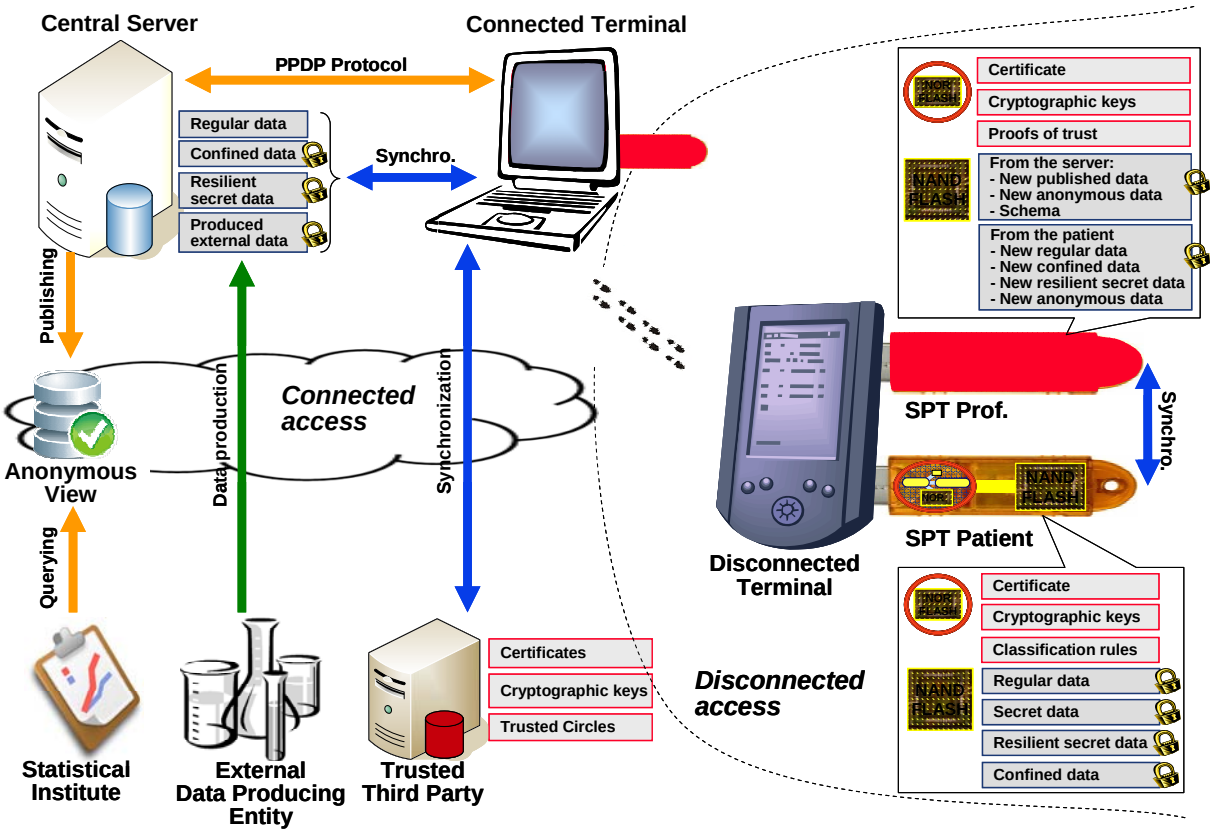

Figure 2. Functional architecture

Figure 2 depicts the global architecture, showing where information resides and whether it is encrypted or not. Data located in dashed rectangles resides in a trusted storage (either the SPT's internal memory or the TTP) contrary to data located in solid rectangles. Data aside yellow locks is encrypted. This architecture provides stronger privacy preservation guarantees than any traditional EHR. Attacks conducted at the server (bypassing the traditional security measures) can only reveal regular data, secret data being absent from the server and confined data being encrypted with keys let under the control of the SPTs and the trusted third party. Attacks conducted over a patient's SPT are made highly difficult by its secure hardware.

4.2. Synchronization

Replicating data on the central server provides availability and durability, but raises a synchronization problem. When a server and a SPT are directly connected with each other, traditional synchronization methods apply. However, a SPT may never connect directly to the central server (e.g. a patient who never leaves home). In this case, SPTs of health professionals must behave as "proxies", carrying encrypted synchronization messages from patients' SPTs to the central server, and vice-versa.

Health professionals carry encrypted synchronization messages from patients' SPTs to the central server when they visit the patients. During the visit, the professional may insert new data in the patient's SPT. At the end of the visit, newly created regular and confined data (i.e., not

present yet in the server) is copied into the professional's SPT. The central server is refreshed every time a professional connects to it. Conversely, the professional SPT carries encrypted data newly created at the server in order to refresh the patient's SPT replica. This situation occurs when external entities produce medical data directly on the central server, e.g., a laboratory producing examination results. However, data cannot be produced in the clear and the data is not yet classified by the patient. To circumvent this problem, external entities must encrypt data with the patient's public key before publishing them on the central server. At synchronization time, the patient will be able to decrypt this data, classify it, and store it according to its privacy class.

4.3. Privacy-Preserving Data Publishing of Data Stored in Spts

Hypothesis on the Anonymity Model

Among the various existing anonymity models, we consider in our context the most popular one, namely the *k*-anonymity model (Samarati & Sweeney, 1998). *k*-anonymity considers that a data row (also called a tuple) is made of three parts: <Identifier, {Quasi-Identifier}, {Sensitive Data}>. An Identifier (ID) identifies precisely an individual (e.g., SSN), Quasi-Identifier (QID) is a set of attributes which, combined with external knowledge, may identify an individual with a high probability (e.g., {age, zipcode}), and Sensitive Data is a set of attributes considered sensitive by the patient (e.g., {diagnosis}). To be *k*-anonymous, a tuple must not contain Identifier and have its Quasi-Identifier indistinguishable from the Quasi-Identifier of at least *k* other tuples. Generalization based algorithms (Sweeney, 1998; Samarati, 2001; Lefevre et al, 2006) aim at generalizing Quasi-Identifiers of an input set of tuples into equivalence classes such that each class contains at least *k* tuples. As an example, consider the schema <SSN, {age, zipcode}, {diagnosis}> and the tuples <1, {105, 75001}, Cancer>, <2, {31, 75002}, Cold>. 2-anonymizing these tuples would yield the following result: <{[31, 105], [75001, 75002]}, Cancer>, <{[31, 105], [75001, 75002]}, Cold>. Hence, an attacker would be unable to infer who has Cancer or Cold even with external knowledge.

The schema of Anonymous Data (i.e., Quasi-Identifiers, and Sensitive Data) depends on the statistical study. It is defined by data analysts, signed by the TTP (because AD specification is sensitive) and the patient must consent to externalize it. Note that for the sake of simplicity, the current version of the protocol does not cope with multiple anonymizations of overlapping Anonymous Data because it may lead to disclosures (Yao et al., 2005; Wang & Fung, 2006; Xiao & Tao, 2007), nor with multiple tuples per individual (Nergiz et al., 2007).

Hypothesis on the PPDP Infrastructure

Computing the equivalence classes requires knowing the whole set of input quasi-identifiers. Due to their unpredictable connection patterns, it is difficult for SPTs to share their QIDs among them. Our PPDP protocol uses the central server to collect QIDs and compute the corresponding equivalence classes, taking care of never disclosing to the server the association between a given QID and a clear-text Sensitive Data. The PPDP protocol does the following assumptions on the infrastructure: (1) participating SPTs are considered seldom connected, powerful enough to perform simple tasks, and highly trustworthy; (2) the server is highly available (24h/7d) and highly powerful; (3) the server is untrusted and can adopt the following attack models:

- *Semi-Honest*: the server obeys the protocol it is participating in but tries to infer confidential data by exploiting in any possible way the result of each step of this protocol;
- *Weakly-Malicious*: the server may cheat the protocol to infer information, but does so only if SPTs can not detect the cheat and if the output remains correct (i.e., no deny of service attacks).

PPDP Protocols

Assuming the existence of an anonymous communication channel between SPTs and the server, a simple way to cope with the Semi-Honest attack model is to use the following three-phases protocol, called *Naive* hereafter:

1. *Collection Phase*: Each contributing SPT sends its QID to the server;
2. *Construction Phase*: When the server has gathered enough QIDs (this judgment can be based on any traditional data utility metric (Fung et al, 2010)), it computes the corresponding equivalence classes;
3. *Anonymization Phase*: The SPT that have contributed to the Collection Phase send their Sensitive Data to the server and tell him to which equivalence class they are associated.

The Naive protocol is secure but it incurs a high – and even unbounded latency – because each participating SPT must connect twice (in the Collection and the Anonymization phases).

By sharing a common encryption/decryption key among SPTs (e.g., inserted at initialization time), the set of SPTs participating to the Collection and the Anonymization phases can be different. Hence, the *Robust* protocol presented below overcomes the latency drawback of Naive:

1. *Collection Phase*: Each contributing SPT sends to the server its QID and its Sensitive Data encrypted with the common key. Note that in order to avoid inferences through encrypted patterns, Sensitive Data must be encrypted in such a way that two identical data do not yield the same encrypted value (e.g., by concatenating a random number to the Sensitive Data);
2. *Construction Phase*: When the server has gathered enough QIDs, it computes the corresponding equivalence classes;
3. *Anonymization Phase*: Every SPT which connects downloads an equivalence class, and returns the decrypted Sensitive Data in background. The time required by this task remains low since it depends on the size of a single equivalence class (i.e., between k and ($2k$-1) tuples, with k usually small).

The Weakly-Malicious attack model assumes that the server can cheat by producing two or more equivalence classes whose associated sets of QIDs overlap. By computing the differences between the k-anonymous overlapping results, the server is able to break the k-anonymity of tuples belonging to the overlapping classes. For example, let C1 and C2 be two overlapping classes, the QIDs belonging to C1 and not to C2 correspond to the tuples appearing in C1 anonymization and not in C2 anonymization, thereby decreasing the effective value of k. We call these cheats *Differential Attacks.* The *WM* protocol builds on the Robust protocol to forbid Differential Attacks. The respective Collection and Anonymization Phases are different in that more information is exchanged in order to allow the detection of attacks. The Construction Phase remains the same. We refer the interested reader to (Allard et al., 2010).

4.4. Use Case

Let us illustrate the behavior of the system through a scenario involving four participants: an elderly patient named Patrick, his family doctor David, a nurse Nora, and a spy Sandra. Patrick, David, and Nora have each their own SPT. Several medical examinations are prescribed to Patrick who classifies them as regular, confined, and secret data. Patrick recently went through blood exams into a medical laboratory. The medical lab performing the examination has published the encrypted results on the central server. Results were encrypted with Patrick's public key obtained from the trusted third party.

Nora frequently visits Patrick at home. Patrick has no internet connection and leaves home seldom. Thus, Nora acts as a synchronization means for Patrick's folder (as do any other person visiting Patrick and owning a SPT). Before the visit, Nora downloads from the central server the latest updates performed in Patrick's folder, encrypted with Patrick's public key. This includes the recent examination results. During the visit, Nora's and Patrick's SPTs synchronize: Nora's SPT send to Patrick's SPT the encrypted examination results, Patrick's SPT decrypts them with his private key, classifies and encrypts them accordingly to their classes – lab results are confined data – and sends them back to Nora's SPT which will refresh the central server the next time it connects to it. Nora's SPT also copies the latest updates performed in Patrick's local folder, if any. Nora cannot get access to this data, protected by the tamper resistance of the SPT.

During a previous visit, Patrick asked David to join his trusted circle. Patrick's SPT hashed and signed David's certificate and this proof of trust was uploaded onto the TTP. After Nora's visit, at his office, he can connect to the central server and view Patrick's up-to-date folder, including the results of the recent examinations (classified as confined data) and possible updates carried back by Nora (in the limit of David's access rights). When visiting Patrick at home, David gets the same information by accessing data directly through Patrick's SPT.

During another visit of David, Patrick's SPT warns Patrick that a nation-wide epidemiological study is being launched and discloses him the schema of the data required by the study. Following David advice, and because Patrick knows that his privacy is well protected, he agrees to contribute. Patrick's SPT forms Patrick's Anonymous data by following the desired schema and runs the collection phase of the PPDP protocol (either Robust or WM depending on the nation-wide policy). Once the construction phase has been completed by the server, every connected SPT can contribute to the anonymization phase of the protocol up to its completion, for the global benefit of all patients.

One day, Patrick's loses his SPT which is found by Sandra. Missing the PIN code, Sandra cannot authenticate to the SPT. She can open the SPT and snoop at the NAND FLASH memory content but data is encrypted. If she tries to tamper the secure chip to obtain the decryption key, security counter measures will destroy the embedded components. The only attack which could be successfully conducted is against the regular data on the central server. Any sensitive data is stored encrypted (the key being within SPTs or on the trusted third party). A few days after, Patrick receives a new SPT, containing both his keys and his data, and continues to receive visits at home from his health practitioners.

5. EXPRESSING PATIENT'S CONSENT

Trusting the EHR security is a prerequisite for the patient to give her consent about a pervasive use of her medical folder but it is definitely not a sufficient condition. Expressing an enlightened consent means understanding and accepting an access control policy specifying who (individuals or roles) is granted access to which part of her folder. This section details the notion of user's privacy and surveys the current models and mechanisms to achieve it.

5.1 Privacy Protection

Legal Approach

Roughly, privacy is the protection of Personally Identifiable Information (PII), by means of restricting access, transfer, storage, etc. of PII. The concept of "informed consent" is a cornerstone of most privacy regulations. The consent of use of personal data must be an enlightened, free, univocal and unilateral act. Protecting PII is a prime concern for the deployment of pervasive computing systems such as EHR (Langheinrich, 2005).

The European Union Directive 95/46/EC sets the protection of individuals with regard to the processing of personal data. Article 29 of this directive establishes a set of core principles of privacy, which are quite close to the ten founding principles of Hippocratic database systems (Agrawal et al., 2002):

1. the *purpose limitation principle*: data must be processed for a specific and declared purpose.
2. the *data quality and proportionality principle*: data must be accurate, adequate and relevant wrt the declared purpose.
3. the *transparency principle*: information must be provided as to the purpose of the processing, the identity of the data controller must be ensured.
4. the *security principle*: appropriate security measures must be taken.
5. the *rights of access, rectification and opposition*.
6. *restrictions on onward transfers*.

As stated by the Article 29 Working party (Article 29 Data Protection Working Party, 2007), one of the essential principles concerning EHR is limiting access to a folder to only those healthcare professionals who are involved in the patient's treatment. Data protection could be enhanced by modular access rights: the patient should be given the chance to prevent access to his EHR data if he so chooses. This requires prior information about the possible consequences of not allowing access.

Privacy Preferences

A lot of attention has been dedicated to expression of privacy preferences, which are consent of use of PII expressed according to the above mentioned principles. The Platform for Privacy Preferences (P3P) defined by the W3C is a machine readable format of privacy preferences. The Platform for Enterprise Privacy Practices (E-P3P) (Karjoth et al., 2002) or EPAL (Ashley et al., 2003) define the enterprise privacy enforcement system for privacy policies internal to the enterprise. P3P is used to state an enterprise's privacy policy when collecting PII from the customers whereas E-P3P is used for internally enforcing the enterprise's policy to control accesses to the collected PII.

As coined by the title of (Massacci & Zannone, 2004) "Privacy Is Linking Permission to Purpose", the purpose (e.g. from P3P: marketing, surveys, payment etc.) is the intended use of the data queried and is the backbone of informed consent. Thus, integrating purpose (and obligations related to these purposes) into control is one of the main challenges and research directions investigated in privacy protection.

5.2. Control Mechanisms for Privacy

Traditional Access Control

An Access Control (AC) (or authorization) policy is a specialized form of security policy, dedicated to permission management. AC primarily aims at enforcing confidentiality (Samarati & Di Vimercati, 2001). Access control is one of the means to enhance privacy by restricting access to personal data. An AC policy is structured according to a model. The model formally describes the language in which policies are expressed and how to decide whether an access request should be granted or denied.
Traditional AC models are the Mandatory Access Control (MAC) and the Discretionary Access Control (DAC). MAC is a label-based (e.g. Unclassified, Confidential, Secret, Top Secret) AC mechanism. Each user and each data is associated to a unique label. Access on a data is allowed if the user is granted sufficient clearance level. DAC is a decentralized user-based mechanism where the creator of a data defines the set of authorizations.

Intermediate concepts between data and users have been introduced to simplify administration of AC policies. In the Role-Based Access Control (RBAC) models family, roles are assigned to users and permissions are assigned to those roles (Ferraiolo et al., 2003). Thus, an RBAC policy is a set of assignments between users and roles and between roles and permissions. The core rule of RBAC states that an access request is granted iff the issuer endorses a role with this privilege.
From the RBAC initiative, several models have been studied in the literature. These models may either extend RBAC (e.g. with temporal or geographical constraints), or organize policies by mean of additional concepts (e.g. team, task, organization) to enhance their expressive power and flexibility. First-order logic has been advocated as a general framework suitable to formalize AC models and policies (Halpern & Weissman, 2008).

Access Control for Privacy

Traditional AC models such as RBAC are commonly used to organize access rights. However, they are not adequate to express finer control on data usage. Usage CONtrol model (UCON) is a foundation for next-generation access control models. In this model, a usage control decision is determined by combining authorizations, obligations, and conditions (Zhang et al., 2005). Usage control is a way to implement digital rights management, for instance by providing guarantees of restriction on onward transfers.

In order to guarantee the purpose limitation principle, the notion of purpose should be used for access control. A policy should ensure that data can only be used for its intended purpose, and the access purpose should be compliant with the data's intended purpose. The authors (Yang et al., 2007) have proposed a purpose-based AC model on this basis. (Ni et al., 2007) have bound purposes to RBAC in an integrated model Privacy-Aware RBAC (P-RBAC). This model has been refined to include the definition of conditional obligations. The integration of purpose control and RBAC for privacy protection of relational data has been investigated too (Byun & Li, 2008).

Specialized Access Control Models for Health Record

Several models have been defined to organize rights on medical data. Alhaqbani and Fidge propose a cascaded AC architecture made of three AC layers (2007). First layer is based on DAC, second one on RBAC and last one on MAC. When all the three policies agree, access is granted. The authors of (Røstad & Nytrø, 2008) have more tightly combined DAC and RBAC into the Personally Controlled Health Record (PCHR) AC system. In their approach, two policies are defined: a common one and personalized one. Only the personalized one is defined by EHR's owner. A conflict resolution rule (e.g. deny overrides, permit overrides) is defined, it is used whenever the two policies disagree on the access decision.

Alhaqbani and Fidge (2007) and Røstad and Nytrø (2008) concentrate on policies defined by the owner of the EHR, data being centralized and held in a single device. Becker and Sewell focus on high level regulations expressed at national level (Becker & Sewell, 2004). They propose a logical language called Cassandra. This language is based on Datalog with constraints, a fragment of first-order logic studied by the databases community which enjoys good decidability properties.

5.3. Toward a Health Record Masking Model

Researches on usage control, purpose-based AC, privacy preferences and practices provide many valuable results to deal with privacy protection and consent expression. Current researches address a broad scope of privacy issues (e.g., enterprise-wide privacy practices or preference language) able to deal with complex rules (e.g., conditional obligations, time restricted usage, logic rules and constraints). However, access control policies usually defined to regulate accesses to EHR systems are far too complex to expect collecting an enlightened consent of the patients on them, as required by the law. This is due to two main characteristics of these policies:

C1. Huge number of AC rules, due to a high number of people interacting with a folder, combined with a large diversity of roles and privileges.
C2. Complexity of the data to be protected, usually described with a highly specialized terminology, and combined with an intrinsic difficulty to determine which data (or data association) reveals a given pathology.

The default access control policy defined for the future French DMP (Personal Medical Folder) illustrates this complexity quite well. As pictured in Figure 3, this RBAC-based policy is expressed as a matrix Document × Role, where elements of Document are the classes of documents constituting a healthcare folder, elements of Role are the roles which can be played by practitioners and each entry gives the corresponding Read and Write privileges. In its current form, this matrix already contains more than 400 entries while classes of documents are very coarse grain to implement an effective control (e.g., radiographies may reveal very different pathologies depending on the organ). What is finally revealed remains obscure to the patient.

In the light of this example, default access control policy must be considered as the expression of the need-to-know principle (i.e., a user should be granted access to the information strictly required to accomplish the tasks related to his role) rather than a tool which can be configured by the patient to better protect his own privacy. The right to hide part of his medical history has however been recognized by the law to the patients, with a prior information about the possible consequences of this action. We believe that collecting the enlightened consent of the patient

requires providing them with effective and comprehensive tools to mask the undesirable information in his folder. To this end, we devised a masking model which consists in defining additional rules, semantically meaningful for each patient (based on self defined terms) and which takes priority over the default access control policy the patient cannot master.

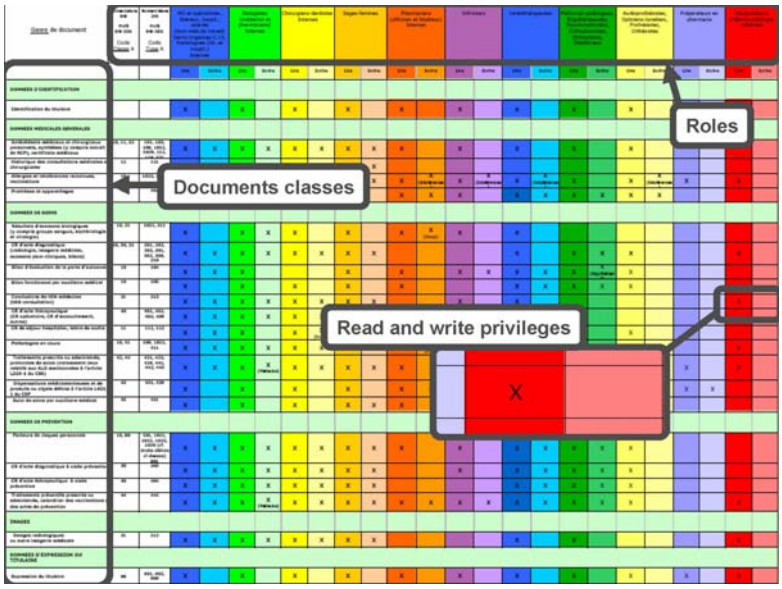

Figure 3. Default matrix of the French DMP (Personal Medical Folder)
http://www.d-m-p.org/docs/TabCxPS.pdf

6. EBAC: AN EVENT-BASED ACCESS CONTROL MODEL

The Event-Based Access Control model (EBAC) has been designed to help the patient in masking sensitive healthcare records in his folder. Design rational of EBAC is simplicity and accuracy. The model is organized according to the main concepts of events, episodes and relation of confidence.

- *Event*: any document added to a medical folder is associated to an event. Events are endowed with properties, among which the *author* of the document (i.e., a practitioner) and the medical *episode* it belongs to.
- *Episode*: an *episode* is a set of events semantically linked and for which the patients wants to define a common masking policy. For example, the patient may define episodes "MyAbortion2008", "MySecondDepression" and associate incoming events to them, potentially with the help of his family doctor. The patient defines his masking policy on an episode basis by defining who (role or identified users) is granted to participate to this episode.
- *Relation of confidence*: the participation of a practitioner P to an episode is regulated by a relation of confidence with the patient stating (1) which event P can see in this episode and (2) who can see the events produced by P himself in this episode (e.g., Dr Guru can see only the events he produces and nobody else than Guru and the patient can see these same events). In other words, the participants of an episode constitute a trusted circle as

defined in Section 4.1 and the relation of confidence defines the scope of their respective actions in this episode. To make the model simple and intuitive, we introduce two scopes for the read the write actions termed *shared* (denoted by S) and *exclusive* (denoted by X). Combining these scopes leads to four possible relations of confidence:
SS: practitioner P can read the shared events in the episode, and produces himself shared events for the episode;
SX: practitioner P can read the shared events in the episode, and produces exclusive events for the episode;
XS: practitioner P can read the exclusive events produced by him in the episode, and produces shared events for the episode;
XX: practitioner P can read the exclusive events produced by him in the episode, and produces exclusive events for the episode;

The main ideas of the model are therefore:

- There exists a default (role-based) AC matrix which is defined at the regulation level and that cannot be modified by the owner of the EHR,
- Each healthcare record is associated to an event, itself related to (at most) one episode and the owner defines his masking policy at the episode level,
- Access decision is taken according to the identity of the querier, the author of the event and the episode the event belongs to, with priority given to the masking rule in case of conflict with the AC matrix,
- Only read permission is considered: the aim on the EBAC model is only to prevent from privacy disclosure, other actions are controlled at the regulation level.

The next section formalizes the EBAC model with sets and functions. Note that we do restrict ourselves to information related to AC decision. In a real implementation, basic types would be refined into more complex types. In the definitions, ∇ denotes the absence of value, $A \times B$ is the Cartesian products of sets A and B, and $\wp(U)$ denotes the set of all subsets of a set U.

6.1. Formal Definitions

Let's introduce basic types:

- *Identifiers*: the set of events' identifiers.
- *Users*: the set of users' identifiers.
- *Form*: the set of documents (or forms) constituting the medical folder.
- *Episodes*: the set of episodes.
- *SS, SX, XS and XX:* four functions that maps each episode to a set of users ($Episodes \rightarrow \wp(Users)$). Moreover $SS(e) \cap SX(e) \cap XS(e) \cap XX(e) = \varnothing$.
- *Events*: the set of events.
- $id : Events \rightarrow Identifiers$, $form : Events \rightarrow Forms$, $author : Events \rightarrow Users$ and $episode : Events \rightarrow Episodes \cup \{\nabla\}$:four functions that maps an event to (respectively) an identifier, a form, an author and an episode (if any).

Note that the sets S_{ep} and S_{∇} defined respectively as $S_{ep} = \{e \in Events \mid episode(e) = ep\}$ and $S_{\nabla} = \{e \in Events \mid episode(e) = \nabla\}$ define a partition of events according to the (unique) episode they belong: $\bigcup_{ep \in Episodes} S_{ep} \cup S_{\nabla} = Events$

Now we introduce two relations for the default access control matrix. Actually, it is role-based, but on more generic treatment it could be any access control model provided that a function $Users \times Event \rightarrow \{true, false\}$ exists. The definitions we propose are related to the flat RBAC model but may be easily extended by role hierarchy. Following definitions are from [Ferraiolo et al. 2003]:

- *Roles*: the set of roles
- $URA \subseteq Users \times Roles$: a relation for "user-role assignment"
- $PRA \subseteq Roles \times Forms$: a relation for "permission-role assignment"
- $DefaultMatrix \subseteq Users \times Forms = \{(u, f) \in Users \times Forms \mid \exists r \in Roles \wedge (u, r) \in URA \wedge (r, f) \in PRA\}$ the relation defined as the join on roles of *URA* and *PRA* relations.
- $defaultAccess :: Users \times Event \rightarrow \{true, false\}$ a function that determine whether a user's access query on an event is granted. The function is defined as:
  - $defaultAccess(u, e) = true$, iff *form(e)=f* and *(u,f)* $\in$ *DefaultMatrix*

Now, we define the semantic of the sets of users SS, SX, XS and XX of a given episode e. The idea is to use these sets to define rights based on the identity of the user who try to access and the identity of the user who wrote the event. A user is always able to access an event he wrote himself.

- $perceive : Episodes \rightarrow \wp(Users), perceive(e) = SX(e) \cup SS(e)$ is the set of users who may read events related to the episode.
- $hidden : Episodes \rightarrow \wp(Users), hidden(e) = XX(e) \cup SX(e)$ is the set of users whose events are hidden to others.
- $access' : Episodes \times Users \times Users \rightarrow \{true, false\}$: the function that tell whether access to an episode *e* by a user *u*, on a event written by user *a* is granted. The function is defined as:
  - $access'(e, u, a) = true$ iff $(a = u) \vee (u \in perceive(e) \wedge a \notin hidden(e))$
- $access : Users \times Events \rightarrow \{true, false\}$: the function that tell whether an access by a user *u* on an event *e* is granted or not. If the event is not related to any episode (∇), access is granted, else, we rely on function *access'*. The function *access* is defined as:
  - $access(u, e) = true$ if $episode(e) = \nabla$,
  - $access(u, e) = true$ if $episode(e) \neq \nabla$ and $access'(episode(e), u, author(e)) = true$,
  - $access(u, e) = false$ otherwise.

Finally, we combine the access decision based on the default access control model, which express the need to know based on role assignment, and on the episode related to the event. Access is granted if both access control decisions agree.

- $granted : Users \times Events \rightarrow \{true, false\}$: the function that combines *defaultAccess* and *access*, it is defined as:
  - $granted(u,e) = true$ iff $defaultAccess(u,e) \wedge access(u,e)$

6.2. Sample Policy

To illustrate the approach, we define a sample EBAC policy. Four professionals named Guru (an adept of alternative medicine), MyPhysician, MyNurse, and AnotherPhysician are acting with the system. The patient we consider has defined two episodes, one for a cancer and another one for an abortion, with the following rules:

- *E1*, "Cancer": *XX(E1)={Guru}, SS(E1)={ MyPhysician, MyNurse }*. This rule states that Guru, MyPhysician and MyNurse constitute the trusted circle for episode E1, in addition to the patient himself. No other user can read events in this episode whatever the default access control policy. MyPhysician and MyNurse share the documents produced in this episode, except those produced by Guru, because the patients wants to hide that he consults Guru for his cancer.
- *E2, "Abortion": SX(E2)={ MyPhysician, AnotherPhysician }, SS(E2)={ MyNurse }* MyPhysician, AnotherPhysician and MyNurse constitute the trusted circle of episode E2. MyNurse, who did practice the abortion, produces shared events. MyPhysician and AnotherPhysician share these events but what they produce themselves is kept invisible to each other. Such a rule may be defined by the patient after consulting MyPhysican about a problem following the abortion, and before consulting AnotherPhysician for a second opinion, whether the patient does not fully trust MyPhysician diagnosis.

Let us define a sample flat RBAC:

- $(Guru, Physician) \in URA$, $(MyPhysician, Physician) \in URA$, $(MyNurse, Nurse) \in URA$
- $(Physician, General) \in PRA$, $(Physician, Treatment) \in PRA$, $(Nurse, General) \in PRA$

The health record is composed of 7 events:

1. $e1 = (General, MyNurse, \nabla)$
2. $e2 = (Treatment, MyPhysician, \nabla)$
3. $e3 = (General, MyPhysician, E1)$
4. $e4 = (Treatment, Guru, E1)$
5. $e5 = (Treatment, MyPhysician, E2)$
6. $e6 = (General, MyPhysician, E2)$
7. $e7 = (General, AnotherPhysician, E2)$

Using function *granted* the following AC matrix can be computed, where T denotes true and F denotes false:

	∇		E1		E2		
	e1	*e2*	*e3*	*e4*	*e5*	*e6*	*e7*
Guru	T	T	F	T	F	F	F
MyPhysician	T	T	T	F	T	T	F
MyNurse	T	F	T	F	F	F	F
AnotherPhys.	T	T	F	F	F	F	T

A proof of concept of the EBAC model has been developed in the Haskell (http://www.haskell.org) functional programming language.

6.3. Implementation Issues

This section sketches how the EBAC model is implemented in the embedded DBMS (see Section 3.2) and how it is used by patients.

A patient can incrementally define episodes (e.g., "Cancer", "Abortion") for his folder. Then, for each event, he chooses:

1. the set of practitioners constituting the trusted circle for this episode;
2. the relation of confidence attached to each.

To implement those access policies in the SPT using the relational model (Codd, 1970), the following relations are added to the database (the view below is simplified for the sake of simplicity):

Event (**Event_id**, *Episode_id*, *User_id*, Write_scope, …)
Episode (**Episode_id**, Label, …)
User (**User_id**, User_name, …)
Privilege (**Privilege_id**, *Episode_id*, *User_id*, read_scope, write_scope, …)

Notations: attributes in bold are primary keys; attributes in italic are foreign keys.

Relation Event logs the events occurring on the medical folder (e.g., insertion of a document), and refers to relations User and Episode. Relation User stores the set of practitioners interacting with the medical folder, and relation Episode stores the set of episodes defined by the patient. The relations of confidence are stored in the relation Privilege with references to relations Episode and User.

At the time of creation of each event, the reference to the user (User_id) originating the event is specified. The reference to the corresponding episode (Episode_id) is created on demand, i.e., at the time of the insertion or later, and the corresponding write scope (write_scope is set to S or X) is filled at that time by querying the relation Privilege.

For example, at creation of event $e3 = (General, MyPhysician, E1)$ of the previous example, event e3 in relation Event refers to the practitioner MyPhysician in relation User and is linked to the episode Cancer. Since the write scope of MyPhysician for episode Cancer is S, the attribute value write_scope in Event is set to S.

At execution time, a query Q issued by a practitioner P on the folder is rewritten to integrate access control. Our implementation requires joining each result of Q with relation Event, and projecting the event tuples on attributes Event.write_scope, Event.user_id and Event.episode_id. Before delivering the tuples to P, access control is checked by applying the following filtering condition:

Event.user_id = current_user OR (Event.write_scope = S AND Event.episode_id IN CC), where CC denotes the set of episodes the current user participates in (i.e., is member of the related trusted circle). This condition is checked by a system query on relation Privileges executed and stored at the time of the connection of the practitioner to the SPT.

Each result tuple qualified on this condition satisfies the access control policy defined by the user. Finally, attributes Event.write_scope, Event.user_id and Event.episode_id are removed before delivering the result tuple to P.

7. FUTURE WORK

The SPT hardware platform is today operational and the main software components described in the preceding sections have been developed and integrated: central server, embedded web server, embedded DBMS and synchronization protocol. The application itself is being developed and will be experimented in the field by the end of 2009 on a population of about 100 elderly patients and 25 practitioners. The ageing of population makes the health monitoring of elderly people at home crucial. In this context, sensitive data has to be shared between all participants of medical-social networks (doctors, nurses, social workers, home help and family circle) with different access rights. The data must be available at the patient's bedside for a better monitoring of their health cares. For this purpose, the Yvelines District in France has decided to carry out an experimental project of Shared Medical-Social Folder (DMSP in French). In the first step, this project targets elderly people from two gerontology networks. At mid-term, it could be extended to other vulnerable people in unstable or handicapped situation.

ALDS (a home care association) has already carried out a "Common Medical Folder" in paper format, which enables professionals and participants from medical-social sectors to write down crucial facts related to the monitoring of elderly people. While the day-to-day use of this paper folder has proved its efficiency, two burning issues were still unresolved:

- *No privacy*: all participants (practitioners but also social workers, home help and family circle) can read all records in the patient's folder while some patients are facing complex human situations (diagnosis of terminal illness, addictions, financial difficulties, etc).
- *No remote access to the folder*: consequently, the folder is not updated consistently and timely, leading to a lesser accurate monitoring.

The objective of this experiment in the field is precisely to demonstrate the accuracy of the proposed technology to tackle these two issues. This project involves INRIA (the French National Research Institute in Computer Sciences), University of Versailles, SANTEOS (a French EHR provider), Gemalto (the smart card world leader), ALDS (a home care association) and COGITEY (a clinic for elderly people).

8. CONCLUSION

EHR projects are being launched in most developed countries. The benefits provided by centralizing the healthcare information in database servers in terms of information quality, availability and protection against failure are unquestionable. Yet, patients are reluctant to abandon the control over highly sensitive data (e.g., data revealing a severe or shameful disease) to a distant server. In addition, the access to the folder is conditioned by the existence of a high speed and secure internet connection at any place and any time.

This paper capitalizes on a new hardware portable device, associating the security of a smart card to the storage capacity of a USB key, to give the control back to the patient over his medical history. We have shown how this device can complement a traditional EHR server (1) to protect and share highly sensitive data among trusted parties and (2) to provide a seamless access to the data even in disconnected mode. From the architectural point of view, the key point is the embedding in a secure chip of the complete software chain usually running on traditional servers: web server, servlets, DBMS and finally the database itself. From the usage point of view, the key point is a new way of personalizing access control policies with minimal assistance of practitioners. While both contributions are orthogonal, their integration in the same infrastructure allows building trustworthy pervasive healthcare folders.

The solution proposed will be experimented in the context of a medical-social network providing medical care and social services at home for elderly people. The expected outcome of this experiment is to demonstrate the effectiveness of the proposed technology with a positive impact on the coordination of medical and social workers and on the acceptance of patients of an electronic usage of their medical history.

ACKNOWLEDGMENTS

This work is partially funded by the Yvelines District Council through the DMSP project and by ANR (the French National Agency for Research) through the PlugDB project. The authors wish to thank Laurent Braconnier and Jean-François Navarre (Yvelines District Council), Philippe Kesmarszky (ALDS), Sophie Lartigue (COGITEY), Morgane Berthelot (SANTEOS), Jean-Jacques Vandewalle (Gemalto) and Karine Zeitouni (University of Versailles) for their active participation in these two projects. Special thanks are also due to Benjamin Nguyen who participated to the definition of the Privacy Preserving Data Publishing protocol.

REFERENCES

Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2002). Hippocratic Databases. In *Proceedings of VLDB'02* (pp. 143-154).

Alhaqbani, B., & Fidge, C. J. (2007). Access Control Requirements for Processing Electronic Health Records Business. In *Proceedings of the Process Management Workshops* (pp. 371-382).

Allard, T., Nguyen, B., & Pucheral, P. (in press). *Safe Anonymization with Smart Tokens*. Retrieved from http://www-smis.inria.fr/Epublications.html

Alliance. (2008, April 28). *The National Alliance for Health Information Technology, on Defining Key Health Information Technology Term*. Retrieved February 17, 2009, from http://www.hhs.gov/healthit/documents/m20080603/10_2_hit_terms.pdf

Anciaux, N., Benzine, M., Bouganim, L., Pucheral, P., & Shasha, D. (2007). GhostDB: Querying visible and hidden data without leaks. In *Proceedings of the ACM SIGMOD Conference* (pp. 677-688).

Anciaux, N., Bobineau, C., Bouganim, L., Pucheral, P., & Valduriez, P. (2001). PicoDBMS: Validation and Experience. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)* (pp. 709-710).

Anciaux, N., Bouganim, L., & Pucheral, P. (2006). Data Confidentiality: to which extent cryptography and secured hardware can help. *Annals of Telecommunications, 61*(3-4), 267-283.

Anciaux, N., Bouganim, L., & Pucheral, P. (2007). Future Trends in Secure Chip Data Management. *IEEE Data Engineering Bulletin*, 30(3), 49-57.

Article 29 Data Protection Working Party. (2007). *Working Document on the processing of personal data relating to health in electronic health records (EHR)* (Tech. Rep. 00323/07/EN WP 131). Brussels, Belgium: European Commission.

Ashley, P., Hada, S., Karjoth, G., Powers, C. & Schunter, M. (2003). *Enterprise Privacy Authorization Language (EPAL 1.2)*. IBM Tivoli Software, IBM Research.

Becker, M. Y., & Sewell, P. (2004). Cassandra: Flexible Trust Management, Applied to Electronic Health Records. *Computer Security Foundations Workshop* (pp. 139-154).

Bloom, B. (1970). Space/time tradeoffs in hash coding with allowable errors. *Communications of the ACM, 13*(7), 422-426.

Brown, S. H., Lincoln, M. J., Groen, P. J., & Kolodner, R. M. (2003). VistA, U.S. Department of Veterans Affairs national scale HIS. *International Journal of Medical Informatics, 69*, 135-156.

Byun, J., & Li, N. (2008). Purpose based access control for privacy protection in relational database systems. *The VLDB Journal, 17*, 603-619.

Carrasco, L. C. (1999). *RDBMS's for Java Cards? What a Senseless Idea!* ISOL Corp.

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM, 13*(6), 377-387.

Computer World. (2003, December). *NASA Sites Hacked*. Retrieved February 17, 2009, from http://www.computerworld.com/securitytopics/security/cybercrime/story/0,10801,88348,00.html

Computer World. (2006, August). *Survey: 81% of U.S. firms lost laptops with sensitive data in the past year*. Retrieved February 17, 2009, from http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9002493&source=NLT_PM&nlid=8

Dahl, M. R. (2006). *Status and perspective of personal health informatics in Denmark*. Aarhus, Denmark: University of Aarhus, Section for Health Informatics, Institute of Public Health. Retrieved February 17, 2009, from http://www.ieee2407.org/files/ws01_mads01.pps

Door, J-P. (2008). *Le dossier médical personnel* (Information Rep. No. 659). Paris: Assemblée Nationale.

eHealth Insider. (2008, January 16). *German doctors say no to centrally stored patient records*. Retrieved February 17, 2009, from http://www.e-health-insider.com/news/3384/.

Eurosmart. (2008, April). *Smart USB Token*. Retrieved February 17, 2009, from http://www.eurosmart.com/images/doc/WorkingGroups/NewFF/Papers/eurosmart_smart_usb_token_wp_april08.pdf

Ferraiolo, D. F., Kuhn, R. D., & Chandramouli, R. (2003). *Role-Based Access Control.* Norwood, MA: Artech House Publishers.

FierceHealthIT news. (2006, August 20). *Massive data loss at HCA*. Retrieved February 17, 2009, from http://www.fiercehealthit.com/story/massive-data-loss-at-hca/2006-08-21

FierceHealthIT news. (2008, September). *GA hospital health data breach due to outsourcing error.* Retrieved February 17, 2009, from http://www.fiercehealthit.com/story/ga-hospital-health-data-breach-due-outsourcing-error/2008-09-28

Fung, B. C. M., Wang, K., Chen, R., & Yu, P. S. (in press). Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*.

Gordon, L. A., Loeb, M. P., Lucyshin, W., & Richardson, R. (2006). *2006 CSI/FBI Computer Crime and Security Survey*. Retrieved February 17, 2009, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2006.pdf

Halpern, J. Y., & Weissman, V. (2008). Using First-Order Logic to Reason about Policies. *ACM Transactions on Information and System Security, 11*, 1-41.

Husek, C. (2008, August). *ELGA: The Electronic Health Record in Austria*. Paper presented at the International Conference of Society for Medical Innovation and Technology, Vienna, Austria.

Internet Engineering Task Force. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. Retrieved February 17, 2009, from http://tools.ietf.org/html/rfc5246

ISO/IEC. (1999). *Integrated Circuit(s) Cards with Contacts – Part 7: Interindustry Commands for Structured Card Query Language (SCQL). Standard ISO/IEC 7816-7*.

Karjoth, G., Schunter, M., & Waidner, M. (2002). Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data. In *Privacy Enhancing Technologies* (pp. 69-84).

Langheinrich, M. (2005). *Personal Privacy in Ubiquitous Computing*. Unpublished doctoral dissertation, ETH Zurich.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2006). Mondrian multidimensional k-anonymity. In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE)*, Atlanta, GA.

Liebert, T. (2008, March). Ongoing concern over Pentagon network attack. *IT News Digest*. Retrieved February 17, 2009, from http://blogs.techrepublic.com.com/tech-news/?p=2098

Massacci, F., & Zannone, N. (2004). Privacy Is Linking Permission to Purpose. In *Proceedings of the Security Protocols Workshop* (pp. 179-191).

MasterCard International. (2002). *MasterCard Open Data Storage Version 2.0. Technical Specifications.* Purchase, NY: Author.

Nergiz, M. E., Atzori, M., & Clifton, C. W. (2007). Hiding the presence of individuals from shared databases. In *Proceedings of ACM SIGMOD*, Vancouver, BC, Canada (pp. 665-676).

Ni, Q., Trombetta, A., Bertino, E., & Lobo, J. (2007). Privacy-aware role based access. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies* (pp. 41-50).

Pedersen, C. D. (2006, September 8). *MedCom - the Danish Healthcare Data Network*. Paper presented at the Meeting of the AGFA.

Pucheral, P., Bouganim, L., Valduriez, P., & Bobineau, C. (2001). PicoDBMS: Scaling down database techniques for the smartcard. *Very Large Data Bases Journal (VLDBJ), 10*(2-3), 120-132.

Pucheral, P., & Yin, S. (2007). *System and Method of Managing Indexation of Flash Memory* (Patent by Gemalto and INRIA No. 07290567.2).

Røstad, L., & Nytrø, O. (2008). Personalized access control for a personally controlled health record. *Proceedings of the 2nd ACM workshop on Computer security architectures* (pp. 9-16).

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE), 13*(6), 1010-1027.

Samarati, P., & di Vimercati, S. D. C. (2000). Access Control: Policies, Models, and Mechanisms. In *Proceedings of the Foundations of Security Analysis and Design on Foundations of Security Analysis and Design 2000 Conference* (pp. 137-196).

Samarati, P., & Sweeney, L. (1998). Generalizing data to provide anonymity when disclosing information. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART PODS*, Seattle, WA (p. 188).

Smart Card Alliance. (2005). *The Taiwan Health Care Smart Card Project.* Retrieved February 17, 2009, from http://www.smartcardalliance.org/resources/pdf/Taiwan_Health_Card_Profile.pdf

Smart Card Alliance. (2006a). *Smart Card Applications in the U.S. Healthcare Industry* (White Paper No. HC-06001). Retrieved February 17, 2009, from http://www.smartcardalliance.org/resources/hc/Smart_Card_Healthcare_Applications_FINAL.pdf

Smart Card Alliance. (2006b). *German Health Card*. Retrieved February 17, 2009, from http://www.smartcardalliance.org/resources/pdf/German_Health_Card.pdf

Sweeney, L. (1998). Datafly: A system for providing anonymity in medical data. In *Proceedings of the IFIP TC11 WG11.3 11th International Conference on Database Security XI: Status and Prospects* (pp. 356-381).

The Financial Times. (2007, September). *Chinese military hacked into Pentagon*. Retrieved February 17, 2009, from http://www.ft.com/cms/s/0/9dba9ba2-5a3b-11dc-9bcd-0000779fd2ac.html

The International Council on Medical & Care Compunetics. (2009, January 23). *Dutch nationwide EHR postponed. Are they in good company?* Retrieved February 17, 2009, from http://articles.icmcc.org/2009/01/23/dutch-ehr-postponed-are-they-in-good-company

The Times. (2008, December 26). *Patients avoid NHS database blunders by keeping cards close to their chest.* Retrieved February 17, 2009, from http://www.timesonline.co.uk/tol/life_and_style/health/article5397883.ece

The Washington Post. (2007, July). *Consultant Breached FBI's Computers*. Retrieved February 17, 2009, from http://www.washingtonpost.com/wp-dyn/content/article/2006/07/05/AR2006070501489_pf.html

Vandewalle, J-J. (2004). Smart Card Research Perspectives. *LNCS Construction and Analysis of Safe, Secure and Interoperable Smart devices.*

Wang, K., & Fung, B. C. M. (2006). Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD Conference*, Philadelphia.

WFTV. (2008, August 14). *Medical Center Patient Records Posted On Internet*. Retrieved February 17, 2009, from http://www.wftv.com/news/17188045/detail.html?taf=orlc

Xiao, X., & Tao, Y. (2007). m-invariance: Towards privacy preserving re-publication of dynamic datasets. In *Proceedings of the ACM SIGMOD Conference,* Beijing, China.

Yang, N., Barringer, H., & Zhang, N. (2007). A Purpose-Based Access Control Model. In *Proceedings of the Symposium in Information Assurance and Security* (pp. 143-148).

Yao, C., Wang, X. S., & Jajodia, S. (2005). Checking for k-anonymity violation by views. In *Proceedings of the 31st Very Large Data Bases (VLDB) Conference,* Trondheim, Norway (pp. 910-921).

Yin, S., Pucheral, P., & Meng, X. (2009). *A Sequential Indexing Scheme for Flash-Based Embedded Systems*. Paper presented at the International Conference on Extending Database Technology (EDBT).

Zhang, X., Parisi-Presicce, F., Sandhu, R., & Park, J. (2005). Formal model and policy specification of usage control. *ACM Transactions on Information and System Security, 8*, 351-387.

---

[1] The term centralization refers to the fact that the data is stored, organized, made available and controlled by database servers, whatever the computer system infrastructure is (either centralized or distributed).
[2] The Danish health minister stopped the EHR effort in 2006 to make the EHR centrally controllable (Dahl, 2006).
[3] PlugDB is a project funded by ANR, the French National Research Agency: http://www-smis.inria.fr/~DMSP